

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ **6.050103 «Програмна інженерія»**

на тему _____ *Web-застосування для вивчення іноземних мов*

_____ *ІІІ-51*

Виконав: студент ІV курсу, групи _____ *Худа Анна Олександрівна*
(прізвище, ім'я, по батькові) _____ (підпис)

Керівник _____ *старший викладач Халус О.А.*
посада, науковий ступінь, вчене звання, прізвище, ініціали _____ (підпис)

**Консультант
з графічної
документації** _____ *ст. викл. Головченко М.М.*
посада, науковий ступінь, вчене звання, прізвище, ініціали _____ (підпис)

Рецензент: _____
_____ (підпис)
посада, науковий ступінь, вчене звання, прізвище, ініціали

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

АНОТАЦІЯ

Пояснювальна записка складається з чотирьох розділів, містить 10 рисунків, 43 таблиці, 10 джерел – загалом 60 сторінок.

Об’єкт дослідження: Web-застосування для вивчення іноземних мов.

Мета дипломного проекту: полегшення та прискорення процесу поповнення словникового запасу користувачеві, що вивчає іноземну мову, за допомогою різноманітних інтерактивних завдань.

У першому розділі виконано аналіз предметної області, відомих технічних рішень, розглянуто аналоги, сформульовано функціональні та нефункціональні вимоги до розроблюваного програмного забезпечення.

У другому розділі було виконано моделювання програмного забезпечення. Побудовано схему бізнес процесів та описано принцип роботи застосування. Також було спроектовано архітектуру програмного забезпечення та обґрунтовано вибір технологій для реалізації застосунку.

У третьому розділі описано специфіку тестування та налагодження програмного забезпечення, описано процеси тестування та наведено декілька контрольних прикладів.

У четвертому розділі описано впровадження програмного забезпечення.

Також наведено: опис програми, технічне завдання, програму та методику тестування, керівництво користувача та графічний матеріал.

Ключові слова: ІНОЗЕМНА МОВА, СЛОВНИКОВИЙ ЗАПАС, ВЕБ-САЙТ, ТЕСТ.

					КПІ.ІП-5123.045440.01.81	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The explanatory note of the of the diploma work consists of 10 pictures, 43 tables, 10 sources – total 60 pages.

The object of study: web-application for learning foreign languages

The aim of diploma work: facilitating and accelerating the vocabulary replenishment process for a user learning a foreign language with a variety of interactive tasks.

In the first section the subject area, known technical solutions and analogues were analysed, also functional and non-functional requirements for the software developed were defined.

In the second section software modeling was performed. The business process diagram was constructed and the application principles were described. Also, the software architecture was designed and the choice of technologies for implementation of the application was substantiated.

In the third section the specifics of testing, debugging software and the testing processes were described. Also, several control examples were provided.

The fourth section describes the implementation of the software.

This work also includes the following materials: a description of the program, a technical task, a program and testing methods, a user manual and graphic material.

Keywords: FOREIGN LANGUAGE, VOCABULARE, WEB-APPLICATION, TEST.

КПІ.ІП-5123.045440.01.81

					КПІ.ІП-5123.045440.01.81	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКРОЧЕНЬ І ТЕРМІНІВ.....	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	12
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ	15
1.3.1 Аналіз відомих технічних рішень	15
1.3.2 Аналіз відомих програмних продуктів.....	15
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	16
1.4.1 Розроблення функціональних вимог	24
1.4.2 Розроблення нефункціональних вимог	29
1.4.3 Постановка комплексу завдань модулю.....	30
1.5 ВИСНОВКИ ПО РОЗДІЛУ	30
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	31
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	31
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	43
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ	51
2.5 ВИСНОВКИ ПО РОЗДІЛУ	52
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	53
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	53
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	53
3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ	55

3.4	Висновок до розділу.....	57
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	58
4.1	Розгортання програмного забезпечення.....	58
4.2	Робота з програмним забезпеченням.....	58
	Висновки.....	59
	Перелік посилань	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Nonce (number that can only be used once) – довільне число використовується під час криптографічного зв'язку лише один раз. Nonce часто випадкове або псевдовипадкове число утворене протоколом аутентифікації, для гарантування унеможливлення використання старих сеансів зв'язку в атаці повторного відтворення [1].

MVC (Model View Controller) - архітектурний шаблон, який застосовують для проектування та розробки програмного забезпечення. Даний архітектурний шаблон ділить систему на три пов'язані між собою частини: модель даних, представлення та контролер. MVC використовується для відокремлення даних та бізнес логіки проекту від представлення, тобто інтерфейсу користувача, так, щоб зміни в представленні не впливали або впливали мінімально на логіку проекту, а зміни в моделі бізнес-логіки могли здійснюватись незалежно від змін інтерфейсу користувача [2].

BPMN (Business Process Model and Notation) - система умовних позначень (нотація) для моделювання бізнес-процесів [3].

CSRF (Cross-Site Request Forgery) – тип веб-атаки, що призводить до виконання певних дій від імені користувача на веб-сторінці де останній аутентифікований. Головною метою CSRF-атаки є підробка запитів, а не викрадення даних, тому що зазвичай зловмисник не має змоги побачити відповідь на підроблений запит. За допомогою соціальної інженерії (наприклад, лист чи повідомлення з посиланням), зловмисник може ініціювати потрібні йому дії на веб-сторінці від імені користувача, наприклад переказ коштів, зміна персональних даних на сайті, пароллю, і т. д. Якщо жертва атаки має адміністративні права на сайті то під загрозою опиняється весь сайт [4].

					КПІ.ІП-5123.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

ВСТУП

У наші дні інформаційні системи є невід'ємною частиною повсякденного життя кожної людини. Щоденно люди користуються комп'ютерами, мобільними телефонами, планшетами та іншими технічними засобами. Також, зараз, у часи, коли люди можуть вільно подорожувати, спілкуватись з людьми з інших країн, а також працювати закордоном, зростає необхідність вивчати іноземні мови, зокерма англійську. Знання однієї або навіть декількох іноземних мов є безсумнівно актуальним для повсякденного та ділового спілкування. Однією із основних проблем, з якою зіштовхуються люди, які почали вивчати або вивчають іноземну мову тривалий час, це – поповнення словникового запасу. Людям, що вивчають іноземні слова, найчастіше не вистачає часу на вивчення слів, або ж не вистачає натхнення заучувати їх.

Мета створення даної роботи – спрощення та прискорення процесу запам'ятовування іноземних слів за допомогою різних інтерактивних завдань.

Завданням даної роботи є розробка застосування для ведення списку слів, виконання користувачем завдань для засвоєння слів, а також для перевірки завдань, виконаних користувачем. Компонентами системи є:

- клієнтська частина WEB-додатку;
- серверна частина WEB-додатку;
- база даних.

Результатом роботи є WEB-застосування, який можна використовувати для спрощення процесу вивчення іноземних слів у будь-якій галузі діяльності.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Були розглянуті деякі відомі методи вивчення іноземних слів. Найефективнішими з них вважаються наступні:

– створення карток зі словами. Звичайні картки із картону є дуже ефективним методом для вивчення іноземних слів. Для використання такого методу необхідні картки вирізані з картону, на яких з однієї сторони пишеться слово на іноземній мові, а з іншої переклад цього слова на рідній мові. З використанням даного методу процес вивчення іноземних слів складається з чотирьох етапів:

1) знайомство зі словами. На цьому етапі необхідно передивлятись картки, проговорюючи слова вголос, та намагатись уявити ці слова та запам'ятати;

2) повторення «іноземна мова – рідна мова». На цьому етапі необхідно дивитись на сторону з іноземним словом та намагатися згадати переклад цього слова на рідній мові. Колоду карток необхідно перетасовувати;

3) повторення «рідна мова – іноземна мова» На цьому етапі необхідно дивитись на сторону зі словом на рідній мові та намагатися згадати переклад цього слова на іноземній мові. Колоду карток необхідно перетасовувати;

4) закріплення. На цьому етапі необхідно якнайшвидше перебрати колоду карток, намагаючись миттєво впізнати слово. Для найкращого результату можна використовувати секундомір і з кожним перебором колоди карт намагатися зменшити час, необхідний на впізнавання карток;

– запам'ятовування слів під час письма. Читання слів – це пасивна мовна діяльність, в той час як написання слів – активна мовна діяльність. Коли

					КПІ.ІП-5123.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

людина пише, словниковий запас розвивається по іншому: людина тренується в написанні вже знайомих слів і переводить їх із пасивного запасу в активний;

- використання зображень. Наш мозок краще запам'ятовує візуальну інформацію;

- написання диктантів. Запам'ятовування нових слів відбувається на рівні механічної пам'яті;

- тестування знань. Час від часу корисно проходити тести не тільки для запам'ятовування слів, а й для розуміння, які слова відклалися у пам'яті, а які – необхідно повторити;

- головне – це час. Науковці, що досліджують роботу людського мозку, стверджують, що необхідно повторювати вивчені слова кожен день хоча б десять хвилин для того, щоб слова залишалися в пам'яті надовго. Після декількох повторень навряд чи доведеться докладати зусиль, щоб згадати його, адже новий словарний запас залишиться назавжди у пам'яті.

Зараз є чимало онлайн ресурсів, що дозволяють вивчати слова техніками, описаними вище не виходячи з дому, однак їх основною проблемою є відсутність можливості користувачу створювати власний словник, що відповідає саме його власним потребам, а там де ця можливість є, вона зазвичай платна.

1.2 Змістовний опис і аналіз предметної області

Для авторизації користувача та надійного зберігання паролів використовується криптографічно-надійна хеш-функція формування ключа bcrypt з використанням nonce (number that can only be used once). При реєстрації користувача, з пароля, який вказав користувач генерується хеш-код за допомогою алгоритму bcrypt з унікальним nonce та зберігається у базі даних. Коли користувач заходить у систему, з пароля, який було введено у формі авторизації, знову генерується хеш-код за допомогою алгоритму bcrypt та порівнюється з хеш-кодом, який зберігається у базі даних для даного

					КПІ.ІП-5123.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

користувача. Якщо хеш-коди однакові, то авторизація проходить успішно, в іншому випадку пароль вважається невірним.

Для генерації кросвордів використовується схема, наведена нижче:

Крок 1. Сортуювання всіх слів по довжині у порядку спадання.

Крок 2. Розмістити перше слово на полі.

Крок 3. Обрати наступне слово.

Крок 4. Перебрати всі слова, що є на полі, і перевірити чи є які-небудь можливі перетини (будь-які спільні літери) з обраним словом.

Крок 5. Якщо є можливе місце для слова, пройтись по усім словам, і перевірити, чи не заважає нове слово.

Крок 6. Якщо слово не заважає, розмістити слово на полі та перейти до кроку 3. В іншому випадку продовжити пошук місця (перейти до кроку 4).

Крок 7. Продовжити цикл до тих пір, доки усі слова не будуть розміщені або не можуть бути розміщені.

Для забезпечення користувача інформацією щодо його рівня знань використовується графік, на якому відображається відсоток вивчених слів та динаміка їх вивчення.

Точки на графіку формуються за такою формулою:

$$\frac{LW * 100}{W}, \quad (1.1)$$

де LW – кількість вивчених користувачем слів,

W – загальна кількість слів користувача.

Слово вважається вивченим коли число правильних відповідей більше за кількість неправильних у 2 рази та загальна кількість правильних більша за 2.

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

В процесі аналізу відомих технічних рішень було виявлено, що сайти, що призначені для полегшення користувачеві процесу запам'ятовування іноземних слів, в основному, використовують картки зі словами. На більшості сайтів на основі цих карток генеруються тестові типи завдань. На деяких сайтах використовуються письмові типи завдань. Також було досліджено, що майже на усіх сайтах до слова додаються зображення, що також полегшує запам'ятовування слів за рахунок використання зорової пам'яті.

Також проводився аналіз відомих алгоритмів генерації кросвордів. В ході дослідження цих алгоритмів було виявлено, що генерація кросворду за допомогою будь-якого алгоритму відбувається на основі порівняння, а отже асимптотична складність алгоритмів генерації кросвордів - $O(n^2)$.

1.3.2 Аналіз відомих програмних продуктів

При аналізі Web-застосувань для вивчення іноземних слів були обрані такі сайти для дослідження: Quizlet, EnglishDom, ProLingvo, WeekEnglish, hosgeldi.com та w2mem.com

Quizlet – один з найпопулярніших ресурсів подібного формату. Відомий у всьому світі завдяки великому вибору мов та хорошій локалізації. Користувач може створювати власні картки зі словами, по яким надалі генеруються різні типи завдань. Наявні такі типи завдань:

- картки – користувач переглядає свої картки зі словами та заучує їх;
- заучування – тестовий тип завдання, в якому користувач повинен обрати переклад/значення даного слова;
- письмо – письмовий тип завдань, в якому користувачу необхідно із літер, що дає система скласти переклад до даного слова;

— правопис — система озвучує слово зі списку слів користувача, а користувач повинен написати це слово;

— тест — завдання, в якому є всі вище перелічені типи завдань.

У платній версії користувач може додавати зображення та передивлюватися графік динаміки вивчення.

EnglishDom — веб-застосування, який також використовує карточки зі словами, але тут користувач вже не може обирати слова для вивчення, а лише додавати слова вже з існуючих списків слів. Після реєстрації користувач відповідає на декілька питань та має змогу пройти тест з англійської для визначення словникового запасу. На основі цього тесту система пропонує користувачеві списки слів для вивчення. Всі вправи тестового типу.

ProLingvo — доволі примітивний сайт, на якому користувачеві дають слова на вивчення, тобто користувач не може обирати набори слів. Завдання є лише одного типу — тестового. Система показує користувачу слова на англійській, користувач повинен запам'ятати переклад цих слів і натиснути кнопку початку тесту. Всі слова перемішуються випадковим чином. Користувач повинен клікнути по слову та його перекладу.

1.4 Аналіз вимог до програмного забезпечення

Користувачами розроблюваної системи можна виділити неавторизованого користувача, авторизованого користувача та адміністратора.

Застосування має забезпечувати такі функції для неавторизованого користувача:

- реєстрація;
- авторизація.

Застосування має забезпечувати такі функції для авторизованого користувача:

- можливість ведення списку слів;
- відображення зображень до слів;

					КПІ.ІП-5123.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

- можливість виконання завдання типу «Кросворд»;
- можливість виконання завдання типу «Написання»;
- можливість виконання завдання типу «Слово»;
- можливість виконання завдання типу «Переклад»
- можливість виконання завдання типу «Зображення»
- можливість перегляду результатів виконання завдань з вказанням помилок;
- вихід із системи.

Застосування має забезпечувати для адміністратора системи функцію додавання словників.

Розроблюване програмне забезпечення повинне надавати наступні варіанти використання:

Таблиця 1.1 – Варіант використання UC-1

Унікальний ідентифікатор	UC-1
Назва	Реєстрація в системі
Опис	Неавторизований користувач реєструється в системі
Учасники	Неавторизований користувач
Передумови	Відкрита сторінка реєстрації нового користувача
Постумови	Користувач зареєструвався в системі
Сценарій	1. Користувач заповнює усі необхідні поля валідними даними; 2. Користувач натискає кнопку реєстрації

Таблиця 1.2 – Варіант використання UC-2

Унікальний ідентифікатор	UC-2
Назва	Авторизація в системі

Продовження таблиці 1.2

Опис	Неавторизований користувач авторизується в системі
Учасники	Неавторизований користувач
Передумови	Користувач зареєстрований в системі. Відкрита сторінка авторизації користувача
Постумови	Користувач успішно авторизувався в системі
Сценарій	1. Користувач заповнює усі необхідні поля валідними даними; 2. Користувач натискає кнопку авторизації.

Таблиця 1.3 – Варіант використання UC-3

Унікальний ідентифікатор	UC-3
Назва	Ведення словників
Опис	Адміністратор додає словник
Учасники	Адміністратор
Передумови	
Постумови	Слова додані до системи
Сценарій	1. Адміністратор обирає у файловій системі файл формату .csv; 2. Словник імпортується в базу даних.

Таблиця 1.4 – Варіант використання UC-4

Унікальний ідентифікатор	UC-4
Назва	Перегляд слів
Опис	Авторизований користувач переглядає список слів
Учасники	Користувач
Передумови	

Продовження таблиці 1.4

Постумови	
Сценарій	1. Користувач переглядає список слів

Таблиця 1.5 – Варіант використання UC-5

Унікальний ідентифікатор	UC-5
Назва	Ведення списку слів
Опис	Авторизований користувач додає слово до списку слів, видаляє зі списку слів або редагує слово
Учасники	Користувач
Передумови	
Постумови	Слово додане до списку слів, видалено зі списку слів або відредаговане
Сценарій	<p>1. Користувач обирає слово зі словника або пише слово вручну на іноземній мові;</p> <p>2. Користувач пише переклад слова;</p> <p>3. За бажанням користувач має змогу додати зображення до слова;</p> <p>4. Користувач натискає кнопку додавання слова.</p> <p>5. Для видалення слова користувач натискає кнопку видалення</p> <p>6. Для редагування слова користувач натискає кнопку редагування та змінює необхідну інформацію у активних полях</p>

Таблиця 1.6 – Варіант використання UC-6

Унікальний ідентифікатор	UC-6
Назва	Перегляд динаміки вивчення

Продовження таблиці 1.6

Опис	Авторизований користувач має можливість переглянути динаміку вивчення слів, як позитивну, так і негативну
Учасники	Користувач
Передумови	Одне чи декілька завдань виконано
Постумови	Сформований графік динаміки вивчення слів
Сценарій	1. Користувач переглядає графік динаміки вивчення слів.

Таблиця 1.7 – Варіант використання UC-7

Унікальний ідентифікатор	UC-7
Назва	Виконання завдання «Кросворд»
Опис	Авторизований користувач виконує завдання з кросвордом, який генерується зі списку слів користувача
Учасники	Авторизований користувач
Передумови	У списку слів користувача 4 чи більше слів для вивчення
Постумови	Результати проходження завдання, зміна графіку динаміки вивчення слів
Сценарій	1. Користувач заповнює згенерований кросворд; 2. Користувач натискає кнопку виконано.

Таблиця 1.8 – Варіант використання UC-8

Унікальний ідентифікатор	UC-8
Назва	Виконання завдання «Написання»

Продовження таблиці 1.8

Опис	Авторизований користувач виконує завдання «Написання», у якому необхідно ввести з клавіатури переклад слова
Учасники	Користувач
Передумови	У списку слів користувача 4 чи більше слів для вивчення
Постумови	Результати проходження завдання, зміна графіку динаміки вивчення слів
Сценарій	1. Користувач вводить з клавіатури переклад слова, яке обрала система.

Таблиця 1.9 – Варіант використання UC-9

Унікальний ідентифікатор	UC-9
Назва	Виконання завдання «Слово»
Опис	Авторизований користувач виконує завдання «Слово», у якому необхідно обрати слово, яке відповідає перекладу обраного системою слова.
Учасники	Користувач
Передумови	У списку слів користувача 4 чи більше слів для вивчення
Постумови	Результати проходження завдання, зміна графіку динаміки вивчення слів
Сценарій	1. Користувач обирає слово, яке відповідає перекладу обраного системою слова.

Таблиця 1.10 – Варіант використання UC-10

Унікальний ідентифікатор	UC-10
--------------------------	-------

					КПІ.ІП-5123.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Продовження таблиці 1.10

Назва	Виконання завдання «Переклад»
Опис	Авторизований користувач виконує завдання «Переклад», у якому необхідно обрати переклад слова, яке відповідає слову рідною мовою, що обрала система
Учасники	Користувач
Передумови	У списку слів користувача 4 чи більше слів для вивчення
Постумови	Результати проходження завдання, зміна графіку динаміки вивчення слів
Сценарій	1. Користувач обирає переклад слова, яке відповідає слову рідною мовою, що обрала система.

Таблиця 1.11 – Варіант використання UC-11

Унікальний ідентифікатор	UC-11
Назва	Виконання завдання «Зображення»
Опис	Авторизований користувач виконує завдання «Зображення», у якому необхідно обрати слово, яке відповідає зображенню, що обрала система
Учасники	Користувач
Передумови	У списку слів користувача 4 чи більше слів для вивчення, до кожного з цих слів є прикріплене зображення
Постумови	Результати проходження завдання, зміна графіку динаміки вивчення слів

Продовження таблиці 1.11

Сценарій	1. Користувач обирає зображення, яке відповідає слову, що обрала система
----------	--

Таблиця 1.12 – Варіант використання УС-12

Унікальний ідентифікатор	УС-12
Назва	Перегляд проценту успіху
Опис	Користувач після проходження завдання має можливість переглянути результат виконання цього завдання, а саме відсоток правильних відповідей
Учасники	Користувач
Передумови	Користувач пройшов будь-яке завдання
Постумови	Відображення результатів проходження завдання
Сценарій	1. Після проходження завдання, користувач переглядає результат.

Таблиця 1.13 – Варіант використання УС-13

Унікальний ідентифікатор	УС-13
Назва	Перегляд помилок
Опис	Користувач після проходження завдання має можливість переглянути помилки, які він здійснив у завданні
Учасники	Користувач
Передумови	Користувач пройшов будь-яке завдання
Постумови	Відображення списку слів, в яких користувач здійснив помилку і які необхідно повторити.

Продовження таблиці

Сценарій	1. Після проходження завдання, користувач переглядає список слів, в яких він здійснив помилку.
----------	--

Таблиця 1.14 – Варіант використання UC-14

Унікальний ідентифікатор	UC-14
Назва	Вихід з системи
Опис	Користувач виходить з системи
Учасники	Користувач
Передумови	Користувач авторизований у системі
Постумови	Відкрита сторінка авторизації
Сценарій	1. Користувач натискає кнопку виходу з системи.

Схема структурна варіантів використання для кожного з акторів зображена у графічному матеріалі.

1.4.1 Розроблення функціональних вимог

Згідно схеми структурної варіантів використання було сформульовано ряд функціональних вимог. Функціональні вимоги наведені у наступних таблицях:

Таблиця 1.15 – Опис функціональної вимоги RQ-1

Унікальний ідентифікатор	RQ-1
Назва	Реєстрація в системі
Опис	Система має надавати незареєстрованому користувачеві можливість зареєструватися у системі заповнивши поля «Login», «Email», «Password» та «Confirm Password» та натиснувши на кнопку «Sign up»

Таблиця 1.16 – Опис функціональної вимоги RQ-2

Унікальний ідентифікатор	RQ-2
Назва	Авторизація в системі
Опис	Система має надавати неавторизованому користувачеві можливість увійти до системи заповнивши поля «Login», «Password» та натиснувши кнопку «Sign in».

Таблиця 1.17 – Опис функціональної вимоги RQ-3

Унікальний ідентифікатор	RQ-3
Назва	Ведення словників
Опис	Адміністратор системи повинен мати можливість додавати словники, що являють собою файл з розширенням .csv. Система повинна імпортувати наповнення файлу в базу даних.

Таблиця 1.18 – Опис функціональної вимоги RQ-4

Унікальний ідентифікатор	RQ-4
Назва	Перегляд слів
Опис	Система повинна надавати користувачеві можливість переглядати, додані ним слова.

Таблиця 1.19 – Опис функціональної вимоги RQ-5

Унікальний ідентифікатор	RQ-5
Назва	Ведення списку слів

Продовження таблиці 1.19

Опис	Система повинна надавати користувачеві можливість додавати нові слова, видаляти додані слова, а також можливість редагувати слова.
------	--

Таблиця 1.20 – Опис функціональної вимоги RQ-6

Унікальний ідентифікатор	RQ-6
Назва	Перегляд динаміки вивчення
Опис	Система повинна надавати користувачеві можливість переглядати графік динаміки вивчення слів.

Таблиця 1.21 – Опис функціональної вимоги RQ-7

Унікальний ідентифікатор	RQ-7
Назва	Генерація завдання «Кросворд»
Опис	Система повинна генерувати кросворд зі слів, які користувач додав до свого списку.

Таблиця 1.22 – Опис функціональної вимоги RQ-8

Унікальний ідентифікатор	RQ-8
Назва	Генерація завдання «Написання»
Опис	Система повинна обирати до 20 випадкових перекладів слів і надавати користувачеві можливість написати переклад цих слів іноземною мовою.

Таблиця 1.23 – Опис функціональної вимоги RQ-9

Унікальний ідентифікатор	RQ-9
Назва	Генерація завдання «Слово»
Опис	Система повинна обирати до 20 випадкових перекладів слів і надавати користувачеві можливість вибору з чотирьох варіантів переклад до кожного з цих слів іноземною мовою.

Таблиця 1.24 – Опис функціональної вимоги RQ-10

Унікальний ідентифікатор	RQ-10
Назва	Генерація завдання «Переклад»
Опис	Система повинна обирати до 20 випадкових слів іноземною мовою і надавати користувачеві можливість вибору з чотирьох варіантів переклад до кожного з цих слів.

Таблиця 1.25 – Опис функціональної вимоги RQ-11

Унікальний ідентифікатор	RQ-11
Назва	Генерація завдання «Зображення»
Опис	Система повинна обирати до 20 випадкових слів іноземною мовою і надавати користувачеві можливість вибору з чотирьох варіантів зображення, яке відповідає кожному з цих слів.

Таблиця 1.26 – Опис функціональної вимоги RQ-12

Унікальний ідентифікатор	RQ-12
Назва	Відображення проценту успіху завдання.

Продовження таблиці 1.26

Опис	Система повинна рахувати та відображати процент успіху, який відповідає проценту правильних відповідей.
------	---

Таблиця 1.27 – Опис функціональної вимоги RQ-13

Унікальний ідентифікатор	RQ-13
Назва	Відображення помилок.
Опис	Після проходження користувачем завдання, система повинна відображати список слів, у яких користувач зробив помилку.

Таблиця 1.28 – Опис функціональної вимоги RQ-14

Унікальний ідентифікатор	RQ-14
Назва	Вихід із системи.
Опис	Система повинна надавати користувачеві можливість вийти з системи та перейти на сторінку авторизації.

Залежності між варіантами використання та функціональними вимогами показано на матриці трасування, що зображена на рисунку 1.1 – Матриця залежності між варіантами використання та функціональними вимогами

	RQ-1	RQ-2	RQ-3	RQ-4	RQ-5	RQ-6	RQ-7	RQ-8	RQ-9	RQ-10	RQ-11	RQ-12	RQ-13	RQ-14
UC-1														
UC-2														
UC-3														
UC-4														
UC-5														
UC-6														
UC-7														
UC-8														
UC-9														
UC-10														
UC-11														
UC-12														
UC-13														
UC-14														

Рисунок 1.1 – Матриця залежності між варіантами використання та функціональними вимогами

1.4.2 Розроблення нефункціональних вимог

Розроблювана система для вивчення іноземних мов є Web-додатком, що автоматизує процес ведення словників з іноземних мов, а також полегшує їх вивчення. Дана система побудована на основі клієнт-серверної архітектури.

Клієнтська частина потребує наявності одного з перелічених Web-браузерів: Mozilla Firefox, Google Chrome, Internet Explorer, Safari, Opera Browser.

Мова додатку: англійська.

Застосування повинне забезпечувати інтуїтивно зрозумілий інтерфейс користувача, швидку навігацію, елементи додатку повинні бути однозначними і зрозумілими. Також застосування має забезпечувати високу продуктивність.

1.4.3 Постановка комплексу завдань модулю

Метою розроблення даного програмного забезпечення є автоматизація процесу ведення словників з іноземних мов, а також полегшення та прискорення їх вивчення за допомогою використання різноманітних типів завдань, що допомагають швидше та надовго запам'ятовувати іноземні слова за рахунок використання різних типів пам'яті користувача.

При аналізі розроблених варіантів використання, функціональних та нефункціональних вимог були сформульовані такі задачі розробки:

- додавання файлів-словників;
- ведення слів;
- генерація тестових та письмових завдань;
- ведення статистичних даних;
- ведення статистичних даних;
- графічне представлення статистичних даних.

1.5 Висновки по розділу

У даному розділі було проведено детальний аналіз та опис предметної області, були розроблені варіанти використання та побудовані функціональні та нефункціональні вимоги до розроблюваної системи. Наведені таблиці варіантів використання та функціональні вимоги, а також матриця залежностей варіантів використання та функціональних вимог допоможуть покращити процес розробки. Були розглянуті та досліджені вже існуючі технічні рішення для вибору ефективних методів та алгоритмів. Після проведення аналізу аналогів, було досліджено, що, вони, як правило, забезпечують мінімальний функціонал в базовій версії, а розширення функціоналу – платне.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для кращого розуміння процесів, які необхідно реалізувати було використано найбільш зрозумілу нотацію опису бізнес-процесів всіх користувачів системи – BPMN. У цьому розділі було змодельовані основні бізнес-процеси, що відбуваються у системі, а саме:

- реєстрація користувача;
- авторизація користувача;
- додавання слова користувачем;
- виконання тестових типів завдань;
- виконання завдання письмового типу;
- виконання завдання типу кросворд;
- додавання файлу-словника адміністратором системи.

Схема структурна бізнес-процесу авторизації наведена на рисунку 2.1 –
Схема структурна бізнес-процесу авторизації.

Користувач на стороні клієнту заповнює поля логіну та паролю та натискає кнопку авторизації. Введені користувачем дані відсилаються на сервер. Сервер звертається до бази даних та отримує запис користувача за логіном, далі сервер обчислює хеш-код від введеного користувачем паролю за допомогою криптографічно-стійкої функції bcrypt та порівнює хеш-код паролю, що зберігається в базі даних. Якщо хеш-коди співпадають, то і пароль введений користувачем співпадає з паролем в базі даних. Якщо паролі співпадають, то користувача авторизовано та відбувається перехід на головну сторінку Web-додатку на стороні клієнта. Якщо ж паролі не співпадають, то на стороні сервера формується повідомлення про помилку, яке відображається на стороні клієнта.

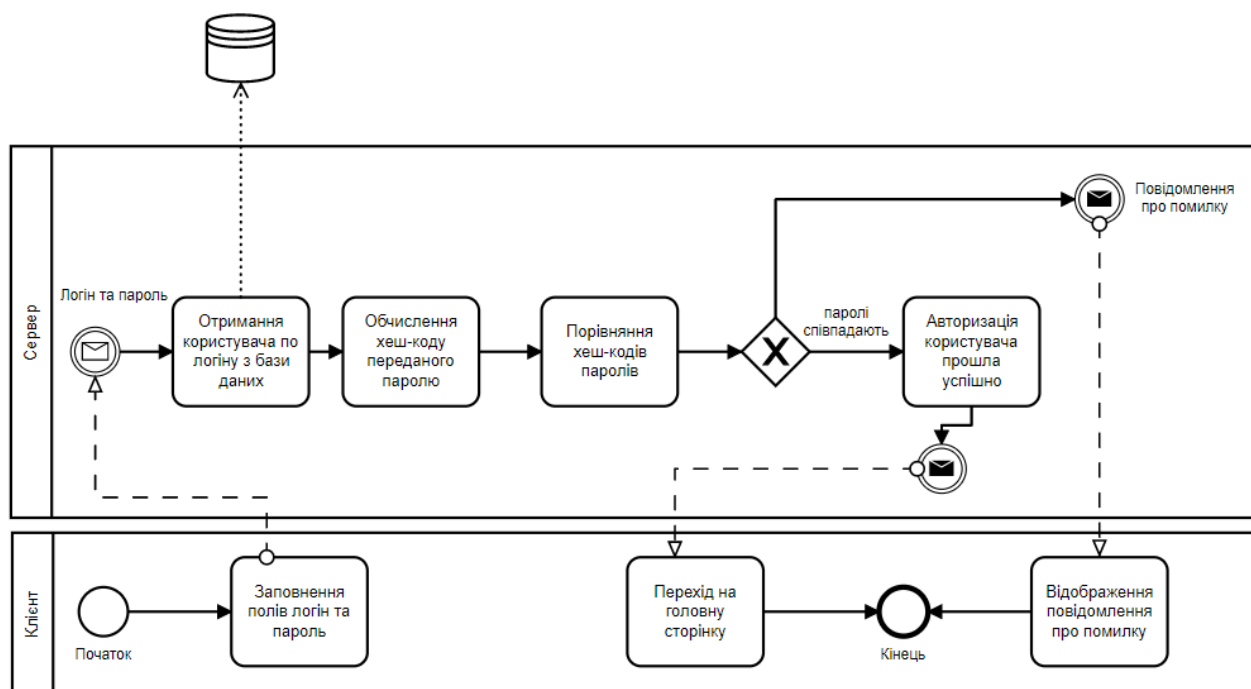


Рисунок 2.1 – Схема структурна бізнес-процесу авторизації

Схема структурна бізнес-процесу реєстрації користувача наведена в графічному матеріалі.

Користувач на стороні клієнта заповнює поля вводу, необхідні для реєстрації, та натискає кнопку реєстрації. Дані, введені користувачем, відправляються на сервер для подальшої обробки. Сервер спочатку перевіряє коректність введених даних. Якщо дані некоректні, користувач бачить помилку на стороні клієнту. Якщо ж з даними все нормально, то сервер звертається до бази даних та перевіряє чи не існує користувача з такими даними. Якщо користувач з такими даними існує, то сервер формує повідомлення про помилку, що відображається на стороні клієнта. Якщо ж користувач унікальний, сервер обчислює хеш-код, введеного користувачем, паролю за допомогою криптографічно-стійкої функції bcrypt та зберігає дані користувача в базу даних. Сервер перенаправляє користувача на сторінку авторизації, що відображається на стороні клієнту.

Схема структурна бізнес-процесу додавання слова наведена на рисунку 2.2 – Схема структурна бізнес-процесу додавання слова.

Користувач на стороні клієнта вводить слово, переклад до цього слова та, за бажанням, додає зображення. Після цього натискає кнопку додавання слова. Дані, введені користувачем, відправляються на сервер для подальшої обробки. Сервер спочатку звертається до бази даних та перевіряє чи немає такого слова у списку слів користувача. Якщо таке слово вже є в базі даних, сервер формує повідомлення про помилку, яке відображається на стороні клієнта. Якщо слова немає у списку слів користувача, сервер перевіряє наявність зображення. Якщо зображення є сервер обробляє його, а саме обрізає до квадратної форми і приводить до потрібного розміру. Далі слово записується до бази даних та відображається у списку слів користувача на стороні клієнта.

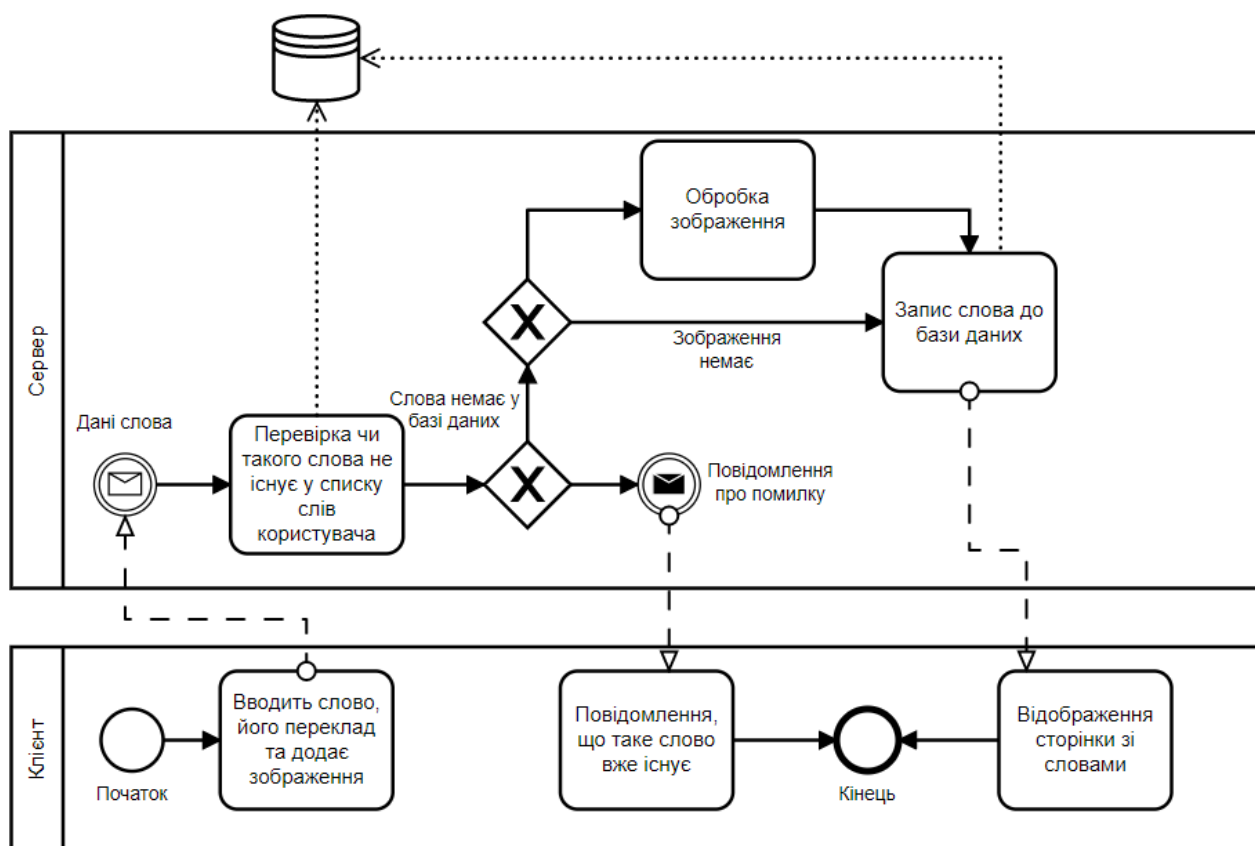


Рисунок 2.2 – Схема структурна бізнес-процесу додавання слова

Схема структурна бізнес-процесу виконання тестових типів завдань наведена на рисунку 2.3 – Схема структурна бізнес процесу виконання

Користувач обирає тип завдання, на сервер відправляється повідомлення з інформацією про вибір користувача. Сервер звертається до бази даних та

перевіряє чи достатньо слів для генерації тестового завдання (4 або більше слова). Якщо слів недостатньо, сервер відправляє повідомлення про помилку, що відображається на сторінці клієнта. Якщо ж слів достатньо, сервер знову звертається до бази даних та обирає до 20-ти випадкових слів, а також 3 випадкові відповіді до кожного слова та перенаправляє на сторінку з питанням, що відображається на сторінці клієнта. Користувач на сторінці клієнта обирає відповідь. Ця відповідь відправляється на сервер, де сервер перевіряє правильність відповіді. Якщо відповідь правильна, кількість правильних відповідей збільшується, в іншому випадку, збільшується кількість неправильних відповідей. Далі сервер перевіряє чи залишились ще слова для тесту. Якщо вони залишились сервер оновлює наступне слово та знову переходить на сторінку з питаннями і так до тих пір, доки слів не залишиться. Після того, як слів більше не залишилось, сервер визначає процент правильних відповідей та оновлює дані динаміки вивчення. На сторінці клієнту відображається сторінка з результатами.

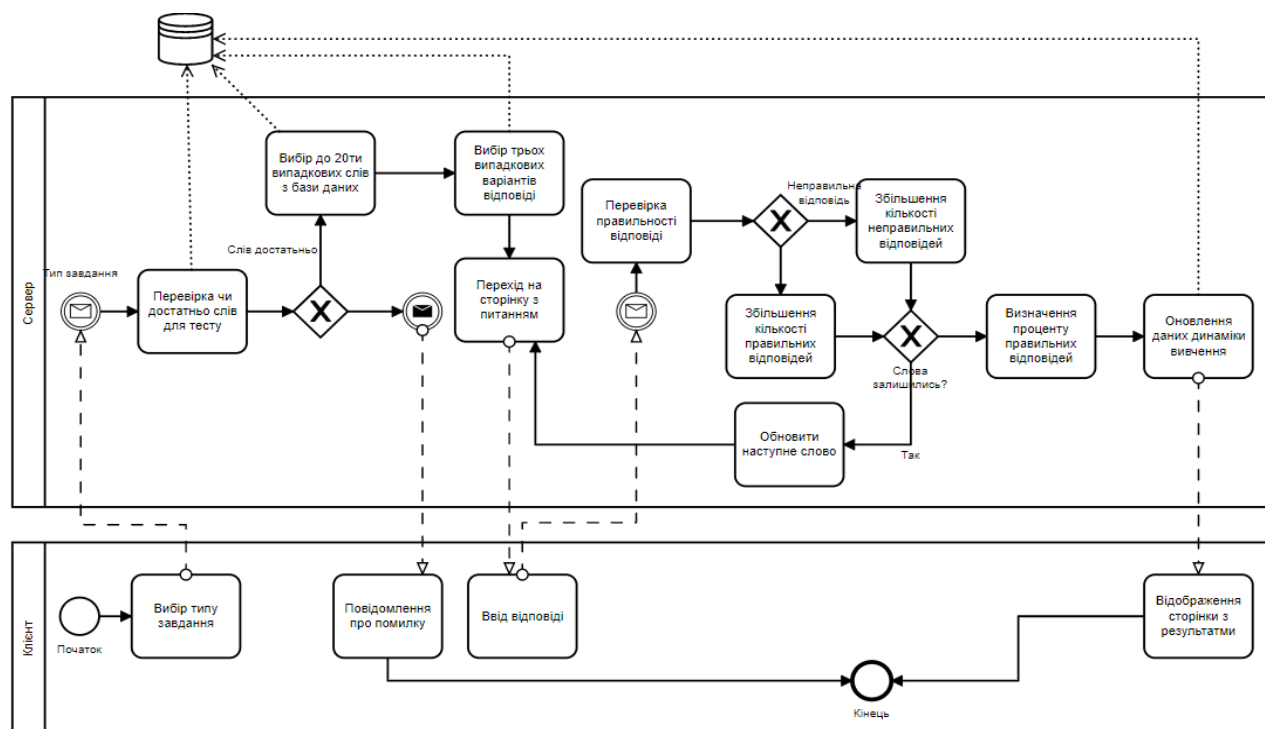


Рисунок 2.3 – Схема структурна бізнес-процесу виконання тестових типів завдань

Схема структурна бізнес-процесу виконання завдання письмового типу наведена на рисунку 2.4 – Схема структурна бізнес-процесу виконання завдання письмового типу.

Користувач обирає тип завдання, на сервер відправляється повідомлення з інформацією про вибір користувача. Сервер звертається до бази даних та перевіряє чи достатньо слів для генерації письмового завдання (4 або більше слова). Якщо слів недостатньо, сервер відправляє повідомлення про помилку, що відображається на стороні клієнта. Якщо ж слів достатньо, сервер знову звертається до бази даних та обирає до 20-ти випадкових слів та перенаправляє на сторінку з питанням, що відображається на стороні клієнта. Користувач на стороні клієнта дає відповідь. Ця відповідь відправляється на сервер, де сервер перевіряє правильність відповіді. Якщо відповідь правильна, кількість правильних відповідей збільшується, в іншому випадку, збільшується кількість неправильних відповідей. Далі сервер перевіряє чи залишились ще слова для тесту. Якщо вони залишились сервер оновляє наступне слово та знову переходить на сторінку з питаннями і так до тих пір, доки слів не залишиться. Після того, як слів більше не залишилось, сервер визначає процент правильних відповідей та оновлює дані динаміки вивчення. На стороні клієнту відображається сторінка з результатами.

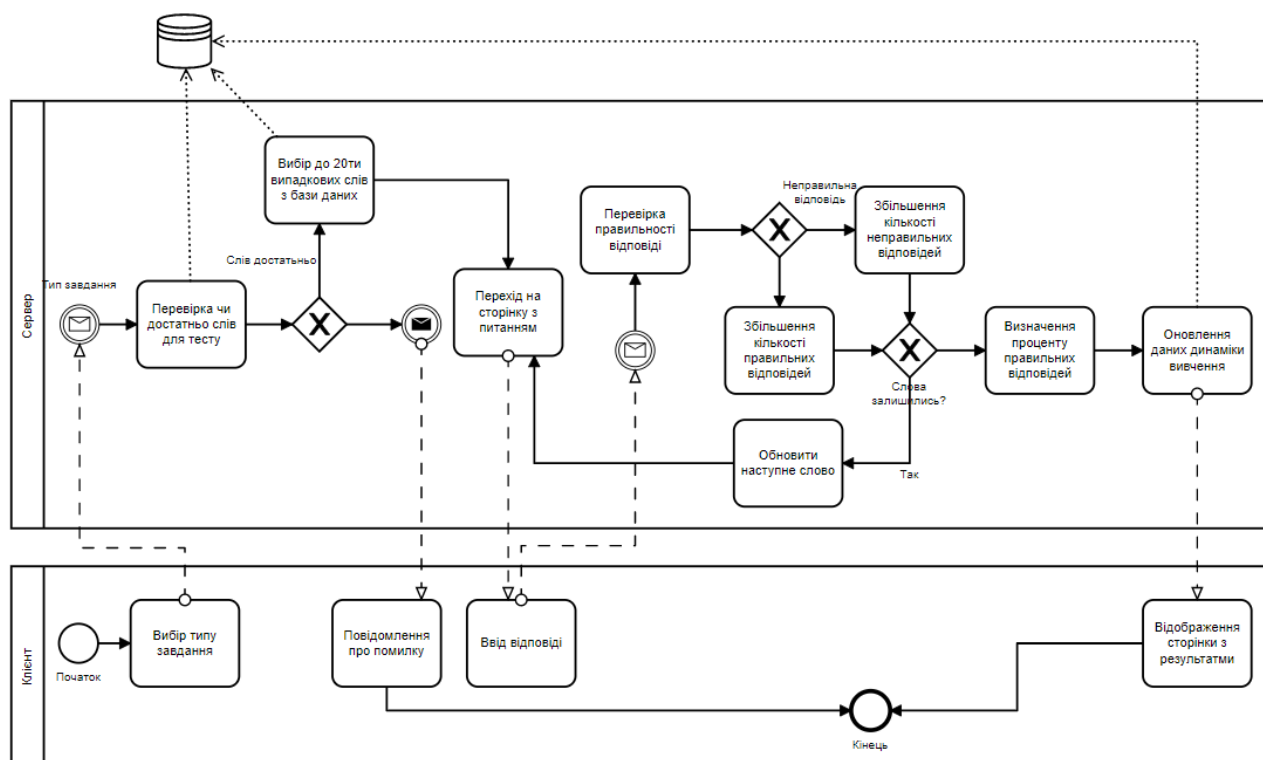


Рисунок 2.4 – Схема структурна виконання завдання письмового типу

2.2 Архітектура програмного забезпечення

Для розробки веб-застосування була обрана мова програмування Java завдяки її властивостям:

- простота – Java володіє лаконічними, тісно пов'язаними один з одним і легко засвоюваними мовними засобами;
- безпека – Java являє собою безпечні засоби для створення веб-застосунків;
- переносимість – програми на Java можуть виконуватись у будь-якому середовищі, де є виконуюча система Java;
- об'єктно-орієнтований характер – в Java втілена сучасна філософія об'єктно-орієнтованого програмування;
- надійність – Java зменшує вірогідність виникнення помилки в програмах завдяки строгій типізації змінних і виконанню відповідних перевірок під час виконання;

- багатопоточність – Java забезпечує вбудовану підтримку багатопоточного програмування;
- архітектурна незалежність – Java не прив’язаний до конкретного типу обсилювальної машини або архітектури операційної системи;
- інтерпретуємість – Java надає байт-код, що забезпечує незалежність від платформи;
- висока продуктивність – байт-код Java максимально оптимізується для підвищення продуктивності;
- розподіленість – Java проектувалася з урахуванням її застосування в розподіленому середовищі Інтернету;
- динамічність – програми на Java містять значну частину інформації, що використовується під час виконання для перевірки і надання дозволу доступу до об’єктів [5].

Застосування побудовано на основі архітектурного шаблону MVC. Схема шаблону MVC наведена на рисунку 2.5 – Структура шаблону MVC.

Контролер управляє запитами користувача. Він обробляє різні типи запитів та викликає відповідні ресурси в моделі.

Модель обробляє дані та логіку застосування.

Представлення відображає дані користувачу в відповідному форматі.

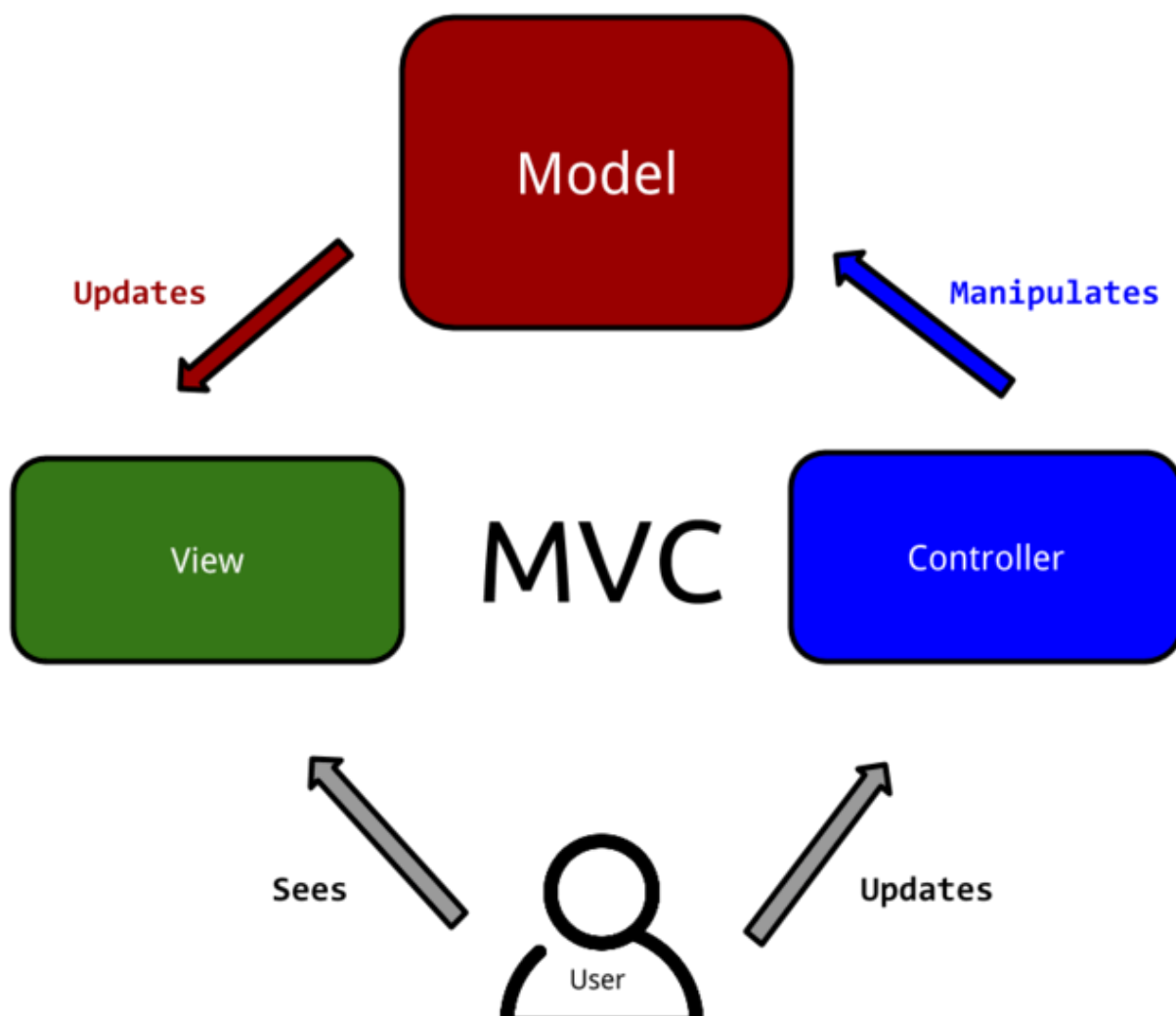


Рисунок 2.5 – Структура шаблону MVC

Spring Framework – універсальний фреймворк, що забезпечує комплексну модель розробки та налаштування для сучасних бізнес-застосування мовою програмування Java. Ключовим елементом Spring є підтримка інфраструктури на рівні застосування, тобто розробники можуть концентруватися на бізнес-логіці застосування без додаткових налаштувань в залежності від середовища виконання [6].

Основними можливостями Spring є:

- впровадження залежностей;
- аспектно-орієнтоване програмування, в тому числі і декларативне управління транзакціями;
- створення Spring MVC-додатків;

— та багато іншого.

Spring можна розглядати як набір інших, менших фреймворків або фреймворків у фреймворку. Більшість цих фреймворків можуть працювати незалежно, але при їх спільному використанні ці фреймворки надають більшу функціональність.

— контейнер інверсія управління: налаштування компонентів застосувань і керування життєвим циклом Java-об'єктів;

— фреймворк аспектно-орієнтованого програмування. Даний фреймворк допомагає працювати з функціональністю, яка не може втілюватися можливостями об'єктно-орієнтованого програмування на Java без втрат;

— фреймворк доступу до даних. Даний фреймворк працює з системами управління базами даних на Java-платформі, за допомогою використання JDBC- і ORM-засобів і забезпечує рішення завдань, які повторюються в великому числі в Java-середовищі;

— фреймворк керування операціями: узгодження різних API керування операціями і інструменти налаштування управління транзакціями для об'єктів Java;

— фреймворк MVC. Даний фреймворк підтримує архітектурний шаблон MVC та реалізує всі основні функції ядра Spring. Детальне зображення схеми взаємодії компонентів Spring MVC представлена на рисунку 2.6 - Схема взаємодії компонентів Spring MVC.

— фреймворк аутентифікації і авторизації: забезпечує інструменти для реалізації та конфігурації процесів аутентифікації та авторизації, що підтримують багато стандартних протоколів, інструментів, практик через дочірній проект Spring Security [6].

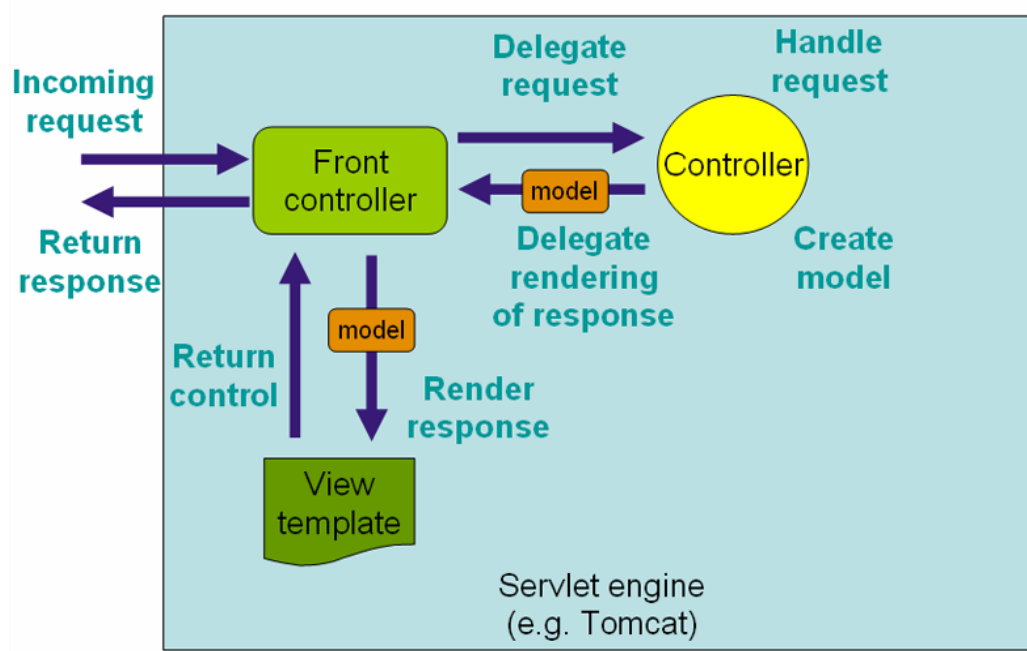


Рисунок 2.6 – Схема взаємодії компонентів Spring MVC

Hibernate - бібліотека для мови програмування Java, що призначена для розв'язання задач об'єктно-реляційного відображення (ORM) [7].

Ціллю Hibernate є полегшення розробнику роботи, яка пов'язана зі значним об'ємом низькорівневого програмування при праці з об'єктно-орієнтованими застосуваннями в реляційних базах даних. Розробники зазвичай використовують Hibernate не лише для полегшення процесу проектування системи класів і таблиць з самого початку, а і для роботи з уже існуючою базою даних [7].

Hibernate вирішує не тільки завдання взаємозв'язків класів Java з таблицями в базі даних (і типів даних Java з типами даних SQL), але і також дозволяє автоматизувати процес створення та оновлення таблиць, виконання запитів до бази даних і обробки результатів цих запитів за допомогою спеціальних засобів. Також цей фреймворк допомагає значно поменшити час розробки застосування, який витрачається на написання SQL- і JDBC-коду самим програмістом. Hibernate генерує SQL-запити автоматично і звільняє програміста від ручного адміністрування отриманого набору даних і

перетворення об'єктів, максимально полегшуючи перенесення застосувань на будь-які бази даних SQL [7].

В якості бази даних обрано PostgreSQL. PostgreSQL - це вільно поширювана об'єктно-реляційна система управління базами даних (ORDBMS), найбільш розвинена з відкритих СУБД у світі і є реальною альтернативою комерційних баз даних [8]. Підтримується на всіх сучасних Unix системах, включаючи найбільш поширені, такі як Linux, FreeBSD, NetBSD, OpenBSD, SunOS, Solaris, DUX, а також під Mac OS X. Починаючи з версії 8.X PostgreSQL працює в "native" режимі під MS Windows.

Основні можливості та функціональність:

– надійність є перевіреним і доведеним фактом і забезпечується наступними можливостями:

- 1) повна відповідність принципам ACID;
- 2) багатоверсійність (Multiversion Concurrency Control, MVCC) використовується для підтримки узгодженості даних в конкурентних умовах, в той час як в традиційних базах даних використовуються блокування;
- 3) наявність Write Ahead Logging (WAL) - загальноприйнятий механізм протоколювання всіх транзакцій, що дозволяє відновити систему після можливих збоїв;
- 4) point in Time Recovery (PITR) - можливість відновлення бази даних (використовуючи WAL) на будь-який момент в минулому, що дозволяє здійснювати безперервне резервне копіювання кластера PostgreSQL;
- 5) реплікація – дані з одного сервера бази даних постійно копіюються на один або кілька інших;
- 6) цілісність даних. Крім MVCC, PostgreSQL підтримує цілісність даних на рівні схеми - це зовнішні ключі (foreign keys), обмеження (constraints);

7) модель розвитку PostgreSQL абсолютно прозора для будь-кого, так як всі плани, проблеми та пріоритети відкрито обговорюються. Користувачі і розробники знаходяться в постійному діалозі через мейлінг листи. Всі пропозиції, патчі проходять ретельне тестування до прийняття їх в програмне дерево. Велика кількість бета-тестерів сприяє тестуванню версії до релізу і вичищення дрібних помилок;

8) відкритість кодів PostgreSQL означає їх абсолютну доступність для будь-кого, а ліберальна BSD ліцензія не накладає ніяких обмежень на використання коду;

– продуктивність PostgreSQL обґрунтована використанням індексів, інтелектуальним планувальником запитів, тонкої системи блокувань, системі управління буферами пам'яті і кешування, чудовою масштабованістю при конкурентній роботі;

– можливість розширення PostgreSQL означає, що користувач може налаштовувати систему за допомогою визначення нових функцій, агрегатів, типів, мов, індексів і операторів. Об'єктно-орієнтованість PostgreSQL дозволяє перенести логіку додатка на рівень бази даних, що сильно спрощує розробку клієнтів, так як вся бізнес логіка знаходиться в базі даних. Функції в PostgreSQL однозначно визначаються назвою, кількістю і типами аргументів;

- багатий набір типів даних;
- простота використання;
- безпечність даних забезпечується 4-ма рівнями безпеки:

1) postgresQL не можна запустити під привілейованим користувачем - системний контекст;

2) SSL, SSH шифрування трафіку між клієнтом і сервером - мережевий контекст;

3) складна система аутентифікації на рівні хоста або IP адреси / підмережі. Система аутентифікації підтримує паролі, зашифровані паролі,

Kerberos, IDENT та інші системи, які можуть підключатися використовуючи механізм підключаємих аутентифікаційних модулів;

4) деталізована система прав доступу до всіх об'єктів бази даних, яка спільно зі схемою, що забезпечує ізоляцію назв об'єктів для кожного користувача, PostgreSQL надає багату і гнучку інфраструктуру.

2.3 Конструювання програмного забезпечення

Як було зазначено раніше архітектура розроблюваної системи базується на шаблоні MVC. На рівні доступу до даних знаходяться Spring Data репозиторії. Вони використовуються сервісами на рівні бізнес логіки, які в свою чергу використовуються контролерами, що обробляють запити користувача і відображають їх у вигляді веб сторінок.

Об'єкти, з якими оперуємо являють собою сутності, що мають пряме представлення в базі даних. Схема структурна бази даних представлена в графічному матеріалі.

Діаграма класів моделей, що відповідає сутностям в базі даних наведена на рисунку 2.7 – Схема структурна класів моделей.

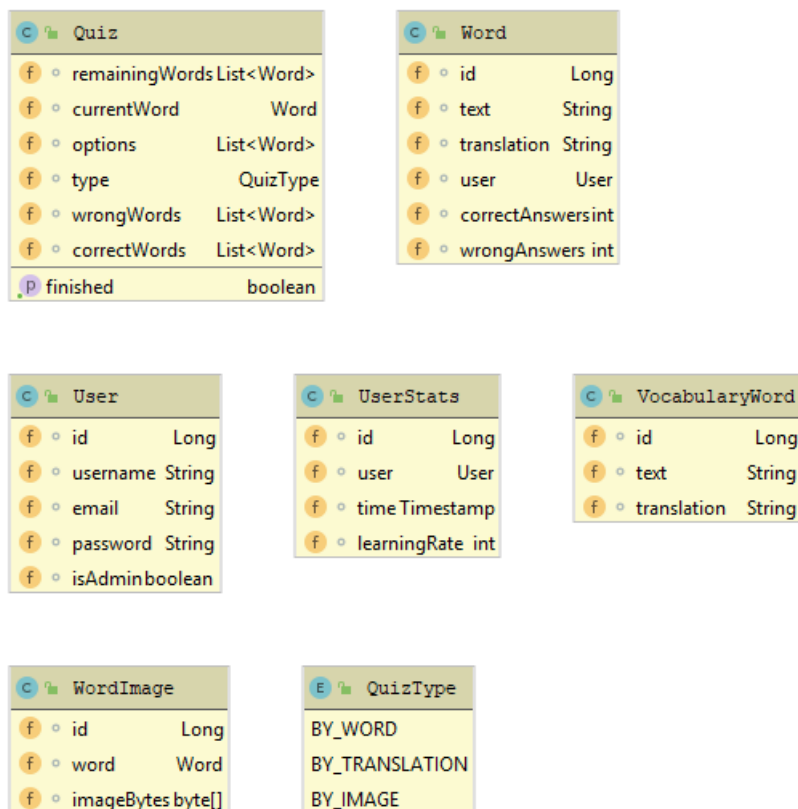


Рисунок 2.7 – Схема структурна класів моделей

Детальний опис класів наведено у наступних таблицях:

Таблиця 2.1 – Опис сутності User

Стовбець	Тип	Первинний ключ	Зовнішній ключ	Опис
id	int	+	-	Унікальний ідентифікатор користувача
Email	Varchar(255)	-	-	Email користувача
is_admin	boolean	-	-	Маркер приналежності користувача до адміністратора

Продовження таблиці 2.1

password	Varchar(255)	-	-	Хеш-код пароллю користувача
username	Varchar(255)	-	-	Логін користувача

Таблиця 2.2 – Опис сутності Word

Стовбець	Тип	Первинний ключ	Зовнішній ключ	Опис
id	int	+	-	Унікальний ідентифікатор слова
correct_answer	int	-	-	Кількість правильних відповідей
text	varchar(255)	-	-	Саме слово
translation	varchar(255)	-	-	Переклад слова
wrong_answers	int	-	-	Кількість неправильних відповідей
User_id	Int	-	+	Унікальний ідентифікатор користувача

Таблиця 2.3 - Опис сутності UserStats

Стовбець	Тип	Первинний ключ	Зовнішній ключ	Опис
id	int	+	-	Унікальний ідентифікатор сутності

Арк.

Продовження таблиці 2.3

learnini_rate	int	-	-	Відсоток вивчених слів
time	timestamp	-	-	Час, коли була зібрана статистика
User_id	int	-	+	Унікальний ідентифікатор користувача

Таблиця 2.4 – Опис сутності WordImage

Стовбець	Тип	Первинний ключ	Зовнішній ключ	Опис
id	int	+	-	Унікальний ідентифікатор зображення
image_bytes	bytearray	-	-	Зображення у вигляді масиву байтів
word_id	timestamp	-	-	Унікальний ідентифікатор слова

Таблиця 2.5 – Опис сутності VocabularyWord

Стовбець	Тип	Первинний ключ	Зовнішній ключ	Опис
id	int	+	-	Унікальний ідентифікатор слова
text	varchar(255)	-	-	Саме слово
translation	varchar(255)	-	-	Переклад слова

Бізнес логіка повністю реалізована в класах сервісів. Структурна схема діаграми інтерфейсів та їх реалізації сервісів зображена на рисунку 2.8 – схема структурна класів сервісів.

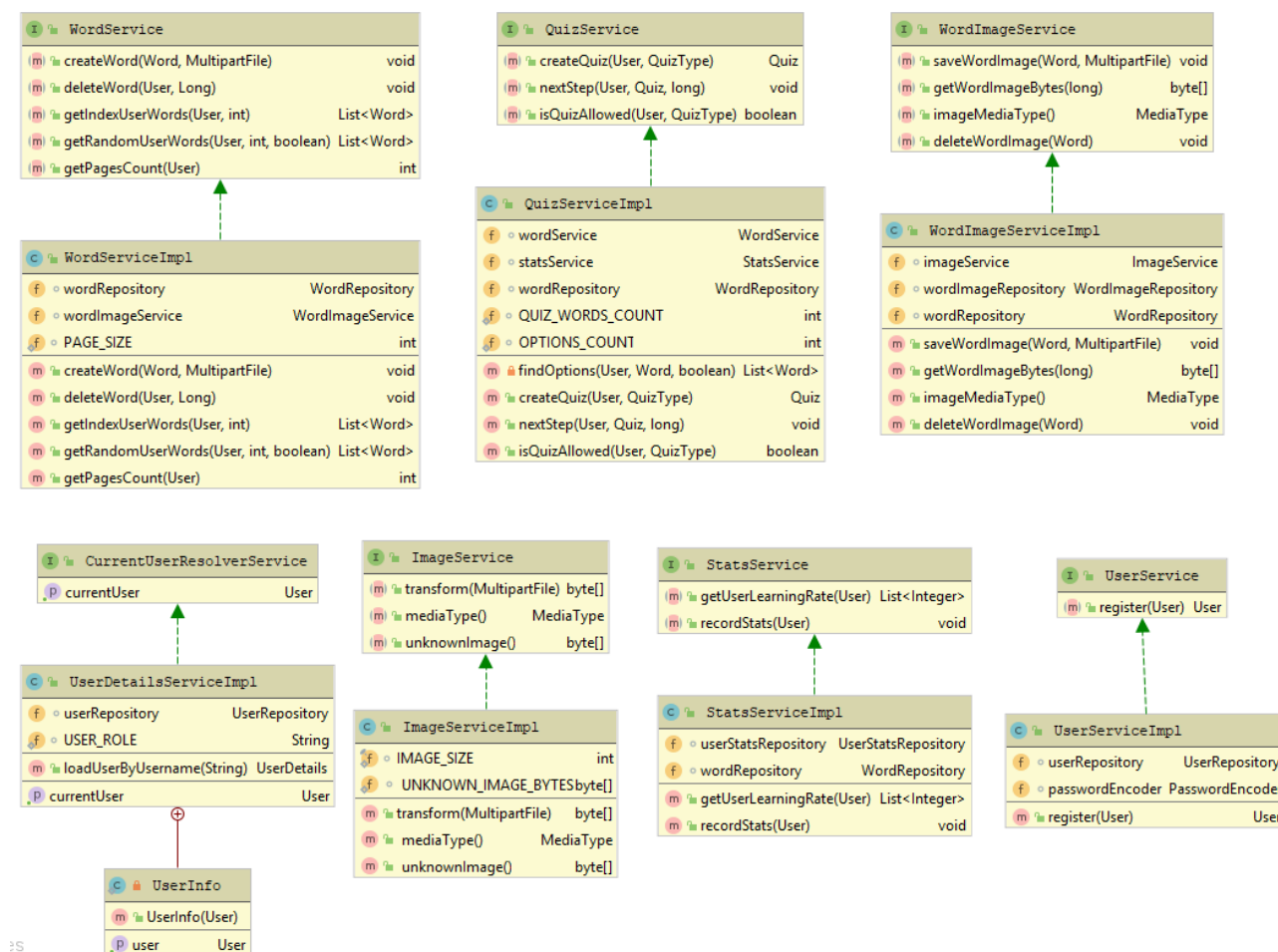


Рисунок 2.8 – Схема структурна класів сервісів

Призначення кожного з сервісів описано у таблиці 2.6 – Призначення сервісів.

Таблиця 2.6 – Призначення сервісів

Назва сервісу	Призначення
QuizService	Даний сервіс призначений для генерації та обробки тестів. В імплементації даного сервісу описана бізнес логіка створення та перевірки завдань

Продовження таблиці 2.6

WordService	Даний сервіс призначений для зберігання слів користувача до бази даних та відображення їх на сторінці користувача
WordImageService	Даний сервіс призначений для зберігання картинки до відповідного слова в базу даних, а також для відображення цієї картинки у списку слів користувача
StatsService	Сервіс, що відповідає за бізнес-логіку формування статистики для конкретного користувача
UserService	Сервіс, що відповідає за реєстрацію користувача
UserDetailsService	Сервіс, що відповідає за зберігання та відтворення інформації про конкретного користувача
ImageService	Сервіс, що відповідає за обробку зображення, а саме обрізання зображення та переведення його до єдиного формату.

Опис методів сервісів описаний у таблицях нижче:

Таблиця 2.7 – Опис сервіса WordService

Метод	Вхідні дані	Вихідні дані	Опис
createWord	Слово, файл з картинкою	-	Валідація слова та збереження в базу даних
deleteWord	Користувач, ідентифікатор слова	-	Видалення слова з бази даних
getIndexUserWords	Користувач, сторінка	Список слів	Отримання списку слів для початкової сторінки

Продовження таблиці 2.7

getRandomUserWords	Користувач, кількість слів, наявність зображення	Список слів	Отримання випадкового списку слів для тесту
getPagesCount	Користувач	Кількість сторінок	Кількість сторінок зі словами на головній сторінці

Таблиця 2.8 – Опис сервісу WordImageService

Метод	Вхідні дані	Вихідні дані	Опис
saveWordImage	Слово, файл з картинкою	-	Додавання зображення до слова
getWordImage	Ідентифікатор слова до зображення	Масив байтів	Отримання зображення до слова
imageMediaType	-	Формат зображення	Отримання формату зображення
deleteWordImage	Слово	-	Видалення зображення

Таблиця 2.9 – Опис сервісу ImageService

Метод	Вхідні дані	Вихідні дані	Опис
transform	Зображення	Масив байтів	Обрізання та трансформація зображення в масив байтів

Продовження таблиці 2.9

mediaType	-	Формат вихідного зображення	Визначення формату вихідного зображення
unknownImage	-	Масив байтів	Отримання зображення слова без картинки

Таблиця 2.10 – Опис сервісу QuizService

Метод	Вхідні дані	Вихідні дані	Опис
createQuiz	Користувач, тип завдання	Завдання	Створення завдання
nextStep	Користувач, тип завдання, ідентифікатор відповіді	-	Перехід до наступного слова
isQuizAllowed	Користувач, тип завдання	Булеве значення	Перевірка чи є можливість створити завдання даного типу

Таблиця 2.11 – Опис сервісу StatsService

Метод	Вхідні дані	Вихідні дані	Опис
getUserLearningRate	Користувач	Масив точок для графіку	Формування точок на графіку по відсотку вивчених слів.

Продовження таблиці 2.11

recordStats	Користувач	-	Формування графіку
-------------	------------	---	--------------------

Таблиця 2.12 – Опис servіcy UserService

Метод	Вхідні дані	Вихідні дані	Опис
register	Користувач	Користувач	Реєстрація користувача

2.4 Аналіз безпеки даних

Для авторизації користувача та надійного зберігання паролів використовується криптографічно-надійна хеш-функція формування ключа bcrypt з використанням nonce (number that can only be used once). При реєстрації користувача, з пароля, який вказав користувач генерується хеш-код за допомогою алгоритму bcrypt з унікальним nonce та зберігається у базі даних. Коли користувач заходить у систему, з пароля, який було введено у формі авторизації, знову генерується хеш-код за допомогою алгоритму bcrypt та порівнюється з хеш-кодом, який зберігається у базі даних для даного користувача. Якщо хеш-коди однакові, то авторизація проходить успішно, в іншому випадку пароль вважається невірним.

Для захисту від CSRF атак використовується CSRF токен для захисту HTTP-запитів, що робить неможливим використання вразливостей такого роду.

Для доступу до Web-сайту використовується протокол HTTPS, що використовує TLS.

2.5 Висновки по розділу

В даному розділі було описано основні технології, засоби та архітектурні шаблони, що використовуються в розроблюваній системі. Була наведена та описана схема бази даних, а також структурна схема діаграми інтерфейсів і структурна схема діаграми класів, що реалізують інтерфейси. Для кожного методу кожного класу було детально описані вхідні та вихідні дані та опис того, для чого потрібен кожен з методів. Було також проаналізовано безпеку системи та виявлено, що паролі в базі даних зберігаються безпечно і що система є стійкою до CSRF-атак.

					КПІ.ІП-5123.045440.01.81	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Якість програмного забезпечення – це сукупність характеристик програмного забезпечення, що відносяться до його можливості задовольняти встановлені та очікувані вимоги. Для оцінки якості програмного забезпечення його необхідно тестувати.

Тестування – це невід’ємна частина процесу розробки будь-якого програмного продукту.

Основними характеристиками якості програмного забезпечення є:

- функціональність – визначається здатністю програмного забезпечення вирішувати задачі, що відповідають потребам користувача;
- надійність – здатність програмного забезпечення виконувати задачі протягом встановленого проміжку часу або вказану кількість операцій;
- зручність використання – можливість легкого розуміння, вивчення, використання та привабливість для користувача;
- ефективність – здатність програмного забезпечення забезпечити необхідний рівень продуктивності;
- портативність – характеризує програмне забезпечення з точки зору легкості його переносу з одного оточення в інше.

Ціллю даного розділу являється опис процесу тестування веб-застосування для вивчення іноземних слів.

3.2 Опис процесів тестування

Об’єктами тестування є:

- сервер по обробці даних, введених користувачем;
- інтерфейс користувача.

Компоненти, що будуть тестуватись:

- реєстрація користувача;
- авторизація користувача;
- ведення списку слів користувача;
- перевірка, виконаних користувачем, завдань на стороні серверу;
- відтворення зображень до слова;
- генерація кросвордів;
- генерація тестових завдань;
- генерація письмових завдань;
- інтерфейс користувача.

Тестування відбуватиметься у режимі білої скриньки, оскільки тестування буде проведено розробником. Для надійності функціонал буде протестований як при позитивних, так і при негативних умовах.

В рамках даного етапу розробки програмного забезпечення будуть проводитись наступні функціональні типи тестів:

- тестування реєстрації;
- тестування авторизації;
- тестування часу відповіді на запити до сервера;
- димне тестування алгоритму генерації кросвордів;
- димне тестування алгоритму генерації тестових завдань;
- димне тестування алгоритму генерації письмових завдань;
- димне тестування перевірки виконання завдань;
- критичне тестування функціоналу по додаванню картинки до слова.

В рамках даного етапу розробки будуть проводитись наступні нефункціональні типи тестів:

- тестування інтерфейсу користувача;
- тестування зручності використання;
- тестування на відмову і відновлення;

- навантажувальне тестування веб-сервера.

Тести будуть виконуватись вручну, оскільки автоматизація тестування для даної розробки не є доцільною тому, що часу на розробку автоматичних тестів недостатньо, а також дане застосування не є великим, що робить час на розробку більшим, ніж час, що витратиться на ручне тестування.

3.3 Опис контрольного прикладу

В процесі тестування була перевірена уся функціональність підсистеми. Послідовно були перевірені всі варіанти використання, результати представлені у відповідних таблицях:

- реєстрація користувача;
- авторизація користувача;
- ведення списку слів;
- відтворення зображень;
- генерація кросвордів;
- генерація тестових завдань;
- генерація письмових завдань;
- відображення графічного представлення статистики.

Опис контрольних прикладів наведений у таблицях:

Таблиця 3.1 – Перевірка реєстрації користувача

Мета тесту	Перевірка реєстрації користувача
Передумови	Користувач з таким логіном та електронною поштою не зареєстрований в системі
Схема виконання	Заповнити всі поля реєстрації, а саме «Login», «Email», «Password» та «Confirm password» валідними даними; натиснути кнопку «Sign up»

Продовження таблиці 3.1

Очікуваний результат	Користувача зареєстровано у системі та перенаправлено на сторінку авторизації
Дійсний результат	Користувача зареєстровано у системі та перенаправлено на сторінку авторизації

Таблиця 3.2 – Перевірка авторизації користувача

Мета тесту	Перевірка авторизації користувача
Передумови	Користувач з таким логіном та електронною поштою вже зареєстрований в системі
Схема виконання	Заповнення всіх полів авторизації, а саме «Login» та «Password» валідними даними; натиснути кнопку «Sign in»
Очікуваний результат	Користувача авторизовано у системі та перенаправлено на головну сторінку веб-застосування
Дійсний результат	Користувача авторизовано у системі та перенаправлено на головну сторінку веб-застосування

Таблиця 3.3 – Перевірка додавання слова з зображенням

Мета тесту	Перевірка додавання слова з зображенням
Передумови	Користувач авторизований у системі

Продовження таблиці 3.3

Схема виконання	У формі додавання слова обрати з файлової системи зображення; написати в полі «Word» або обрати з випадаючого списку слово іноземною мовою; в полі «Translation» написати переклад до цього слова; натиснути кнопку «Add»
Очікуваний результат	Слово додано до списку слів користувача та відображається у списку слів з зображенням до цього слова
Дійсний результат	Слово додано до списку слів користувача та відображається у списку слів з зображенням до цього слова

3.4 Висновок до розділу

В даному розділі були розглянуті основні показники якості та в чому полягає визначення якості програмного забезпечення. Були описані методики, які будуть використані при тестуванні програмного забезпечення.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Схема розгортання компонентів системи представлена на рисунку 4.1 – Схема структурна розгортання компонентів системи.

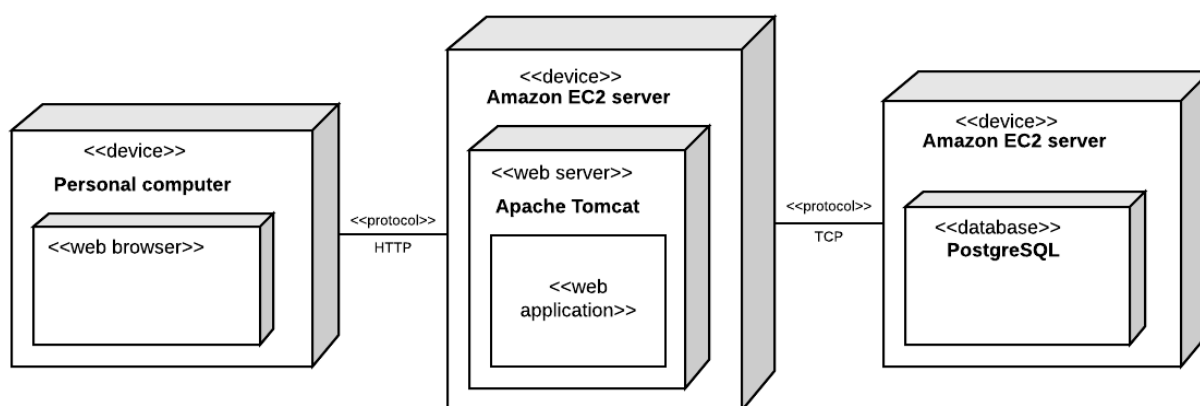


Рисунок 4.1 – Схема структурна розгортання компонентів системи

Для розгортання серверної частини програмного застосування повинні бути встановлені Java 8+ та PostgreSQL 11. Передбачається, що в якості серверів будуть використані сервера сервісу Amazon EC2. Веб-сервер Apache Tomcat є вбудованим в JAR файл застосунку.

Дане програмне застосування є Web-застосуванням, а тому для розгортання клієнтської частини на машині користувача повинен бути встановлений веб-браузер.

4.2 Робота з програмним забезпеченням

Детальна інструкція користувача наведена у Керівництві користувача.

ВИСНОВКИ

При виконанні даного дипломного проекту було детально проаналізовано предметну область, а також були проаналізовані вже відомі технічні рішення та найбільш популярні аналоги. Після виконання аналізу предметної області та наявних аналогів було з'ясовано, що необхідність розробки веб-застосування дійсно існує. Окрім, цього були спроектовані головні вимоги до застосування та функціоналу, що наявний у застосуванні.

Було спроектовану архітектуру, що є масштабованою та легкою у супроводженні. Для розробки додатку використовувалася мова програмування Java.

У якості бази даних була використана реляційна база даних PostgreSQL.

Після розробки додатку було проведено його тестування та доведено, що якість додатку відповідає тій, що від нього вимагалася.

Окрім, цього були описані процеси розгортання додатку та його супровід.

Створене веб-застосування автоматизує процес ведення словників і полегшує та прискорює процес вивчення слів. Дане програмне забезпечення може використовуватись у будь-якій сфері діяльності, де необхідне знання англійської мови.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Криптографічний нонс [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Криптографічний_нонс
- 2) Модель-представлення-контролер [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Модель-вид-контролер>
- 3) BPMN [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/BPMN>
- 4) Міжсайтова підробка запиту [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Міжсайтова_підробка_запиту
- 5) Герберт Шилдт Java 8 Руководство для начинающих, 6-е издание – М: Издательство Вильямс, 2015 – 720 с.
- 6) Крейг Уоллс Spring в действии – М: ДМК Пресс, 2013 – 752 с.
- 7) Christian Bauer, Gavin King Hibernate in action – М: Manning Publications, 2005 – 408 p.
- 8) PostgreSQL [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/PostgreSQL>
- 9) Spring Framework [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Spring_Framework
- 10) Hibernate [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Hibernate>

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

Web-застосування для
вивчення іноземних мов

Технічне завдання
КПІ.ІП-5123.045440.02.91

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.А. Халус

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ А. О. Худа

Київ – 2019 року

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
4.1	ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК.....	6
4.2	ВИМОГИ ДО НАДІЙНОСТІ	7
4.3	УМОВИ ЕКСПЛУАТАЦІЇ	7
4.4	ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ	7
4.5	ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ	8
4.6	ВИМОГИ ДО МАРКУВАННЯ ТА ПАКУВАННЯ.....	8
4.7	ВИМОГИ ДО ТРАНСПОРТУВАННЯ ТА ЗБЕРІГАННЯ	8
4.8	СПЕЦІАЛЬНІ ВИМОГИ.....	8
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	9
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	10
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	11

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Web-застосування для вивчення іноземних слів

Галузь застосування: наведене технічне завдання поширюється на розробку Web-застосування для вивчення іноземних мов, котра використовується для полегшення та прискорення процесу вивчення слів та призначена для будь-яких галузей діяльності, де необхідне знання іноземних мов.

					КПІ.ІП-5123.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки Web-застосування є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5123.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для полегшення та прискорення процесу поповнення словникового запасу користувачеві, що вивчає іноземну мову, за допомогою різноманітних інтерактивних завдань.

Метою розробки є забезпечення користувача можливістю ведення списку іншомовних слів та забезпечення генерації завдань для вивчення цих слів за допомогою впливу на різні типи пам'яті – зорову та механічну.

					КПІ.ІП-5123.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

Для користувача:

- можливість ведення списку слів;
- відображення зображень до слів;
- можливість виконання завдання типу «Кросворд»;
- можливість виконання завдання типу «Написання»;
- можливість виконання завдання типу «Слово»;
- можливість виконання завдання типу «Переклад»
- можливість виконання завдання типу «Зображення»
- можливість перегляду результатів виконання завдань з вказанням помилок;

- можливість перегляду графіку динаміки вивчення слів;
- вихід із системи.

Для адміністратора системи:

- можливість додавання словників.

4.1.2 Розробку виконати на платформі Java.

4.1.3 Додаткові вимоги:

Додаток має забезпечувати такі функції для неавторизованого користувача:

- реєстрація;
- авторизація.

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Забезпечити цілісність інформації в базі даних.

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.2 Обслуговування

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

4.4.2 Мінімальна конфігурація технічних засобів:

Тип процесору: Pentium.

Об'єм ОЗП: 2 Гб.

Операційна система: Windows, Linux, MacOS.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства WIN32 (Windows'XP, Windows NT і т.д.) або Unix.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі: HTTP запити.

4.5.3 Результати повинні бути представлені в наступному форматі: Web-сторінка.

4.5.4 Програмне забезпечення повинно бути доступним через HTTP API.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІП-5123.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 60 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Опис програми

5.3.4 Програма та методика тестування

5.3.5 Керівництво користувача.

5.3.6 Керівництво адміністратора

5.4 Графічна частина повинна бути виконана на чотирьох аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема структурна варіантів використання

5.4.2 Схема бази даних

5.4.3 Схема структурна бізнес-процесу реєстрації користувача

5.4.4 Креслення вигляду екранних форм

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1	Вивчення літератури за тематикою проекту	20.03.2019	
2	Розробка технічного завдання	25.03.2019	Технічне завдання
3	Аналіз вимог та уточнення специфікацій	05.04.2019	Специфікації програмного забезпечення
4	Проектування структури програмного забезпечення, проектування компонентів	10.05.2019	Схема структурна програмного забезпечення та специфікація компонентів
5	Програмна реалізація програмного забезпечення	24.05.2019	Тексти програмного забезпечення
6	Тестування програмного забезпечення	25.05.2019	Тести, результати тестування
7	Розробка матеріалів текстової частини проекту	27.05.2019	Пояснювальна записка
8	Розробка матеріалів графічної частини проекту	07.06.2019	Графічний матеріал проекту
9	Оформлення технічної документації проекту	10.06.2019	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до «Програми та методики тестування».

					КПІ.ІП-5123.045440.02.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

Web-застосування для вивчення іноземних мов

Програма та методика тестування

КПІ.ІП-5123.045440.03.51

“ПОГОДЖЕНО”

Керівник проекту:

_____ О.А. Халус

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ А.О. Худа

Київ – 2019 року

ЗМІСТ

1	ОБ'ЄКТ ВИПРОБУВАНЬ	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ	6

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Web-застосування для вивчення іноземних мов, написаний мовою програмування Java та побудований на архітектурному шаблоні MVC.

					КПІ.ІП-5123.045440.03.51	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

2 МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- процес реєстрації та авторизації;
- функціональна працездатність елементів сторінок web-застосування;
- генерація письмових та тестових завдань;
- забезпечення належного рівня безпеки даних;
- зручність роботи з web-застосуванням;
- відповідність дизайну вимогам Технічного завдання.

					КПІ.ІП-5123.045440.03.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

3 МЕТОДИ ТЕСТУВАННЯ

Тестування виконується методом Gray Box Testing. Перевіряється як код, так і безпосередньо програмний продукт на відповідність функціональним вимогам. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- функціональне тестування, зокрема на рівні димного тестування (тестування базового функціоналу);
- тестування продуктивності програмного забезпечення, зокрема тестування стабільності та навантажувальне тестування;
- тестування API;
- тестування користувацького інтерфейсу.

					КПІ.ІП-5123.045440.03.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

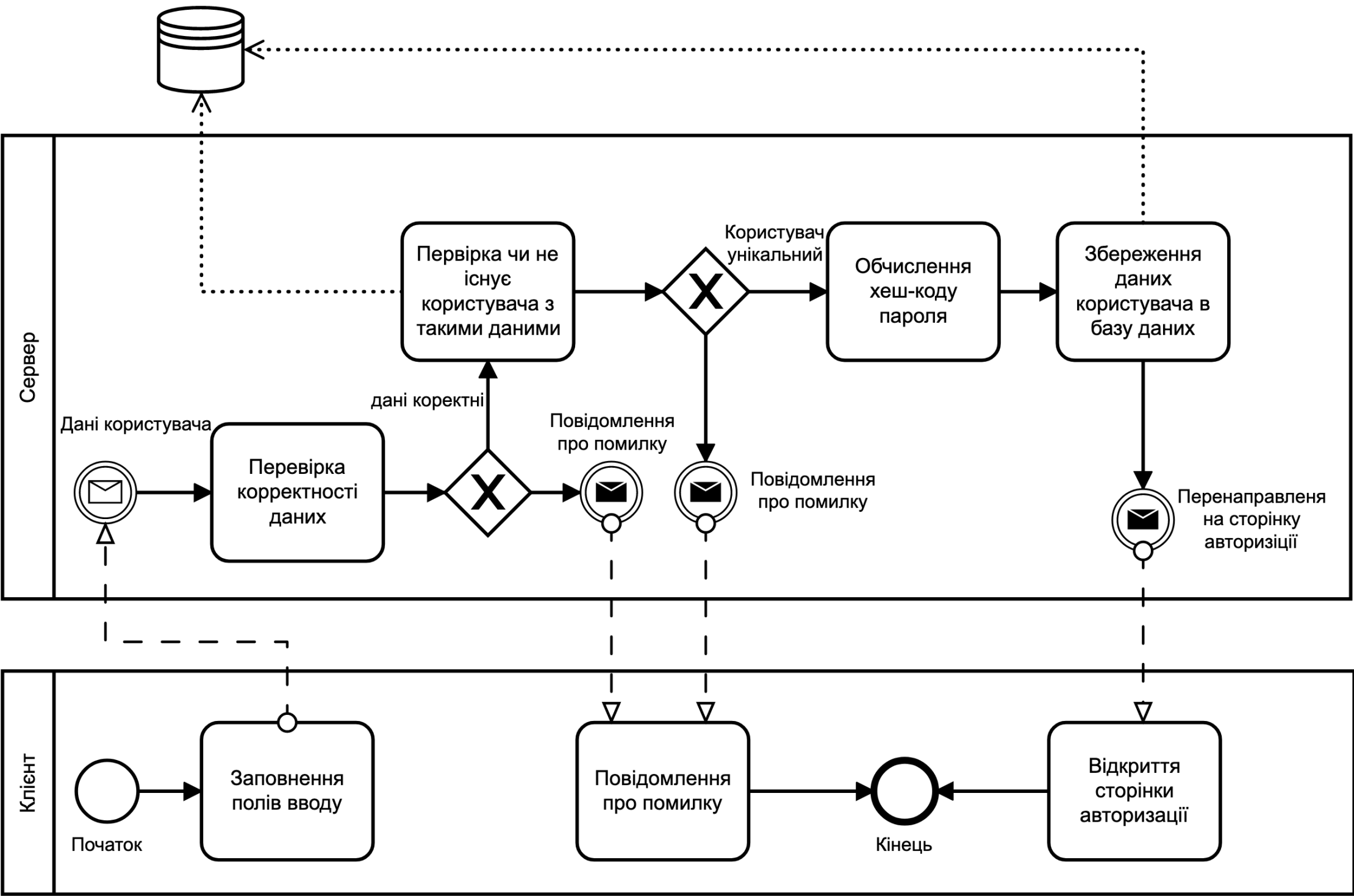
4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Димне тестування виконується вручну. Навантажувальне тестування проводиться за допомогою інструментарію JMeter. Тестування API проводиться з використанням інструментарію Postman.

Працездатність web-застосування перевіряється шляхом:

- ручного тестування – введенням граничних значень класів еквівалентності в поля, які можна редагувати;
- ручного тестування на відповідність функціональним вимогам;
- статичного тестування коду;
- тестування web-ресурсу в різних web-браузерах;
- тестування при максимальному навантаженні;
- тестування стабільності роботи при різних умовах;
- тестування зручності використання;
- тестування інтерфейсу.

					КПІ.ІП-5123.045440.03.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6



					КПІ.ІП-5123.045440.06.99 СС			
					Схема структурна варіантів використання	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Худа А. О.						
Перевірів		Халус О. А.						
Т. кон.					Web-застосування для вивчення іноземних мов	Аркуш		Аркушів
Н. кон.		Ліщук К.І.				КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51		
Затвердив		Халус О. А.						



Login

Password

Sign in

Sing up



Login

Email

Password

Confirm password

Sign up

Sing in

Your dictionary

Choose image	Browse	Word	Translation	Add
	book	книга	Remove	Edit
	crossword	кросворд	Remove	Edit
	diagram	діаграма	Remove	Edit
	friends	друзі	Remove	Edit
Previous 1 Next				

книга

book

friends

crossword

diagram

1

1



Quiz finished!

Your result is:

50%

2

2

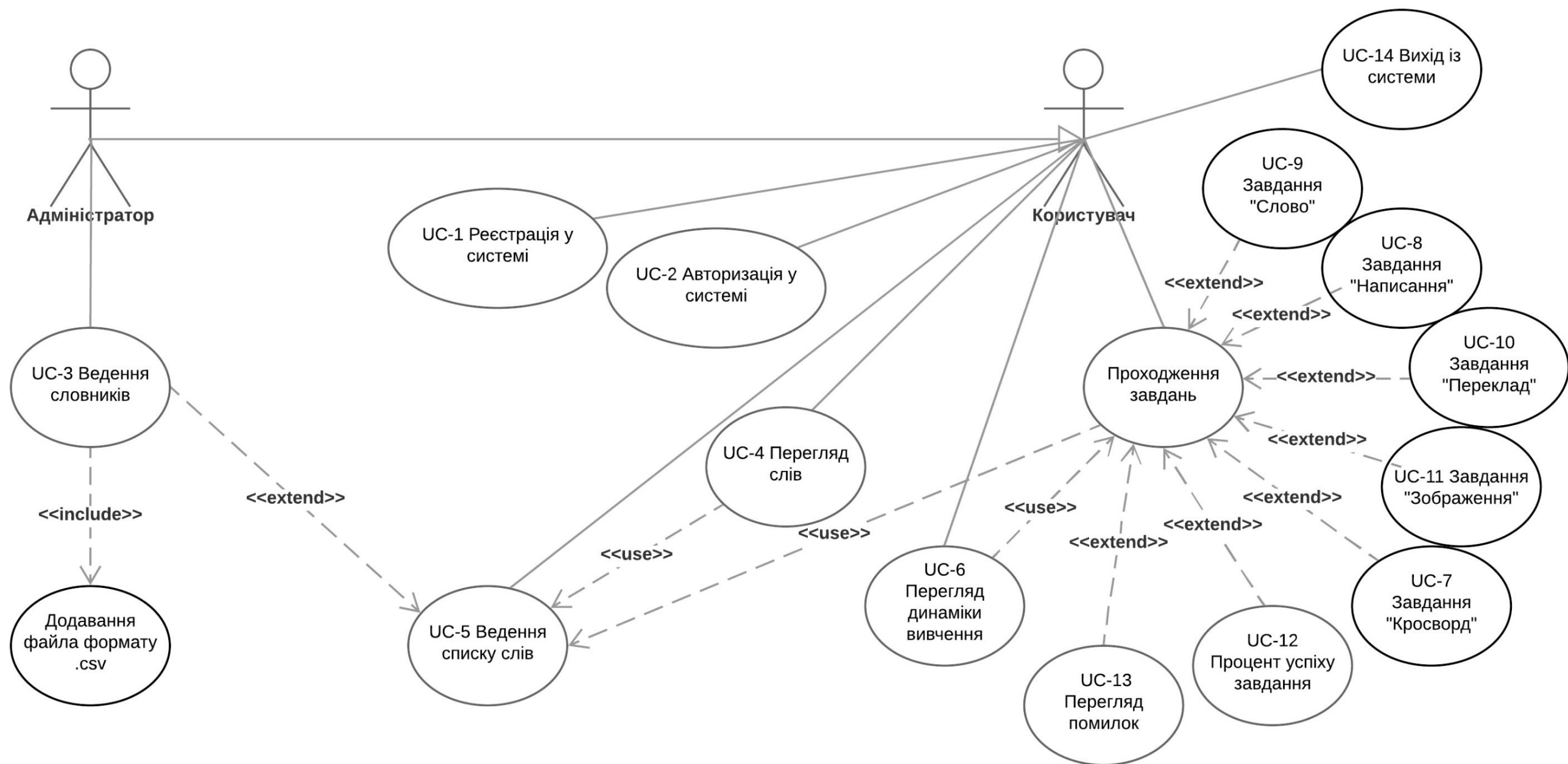
Try again

Home

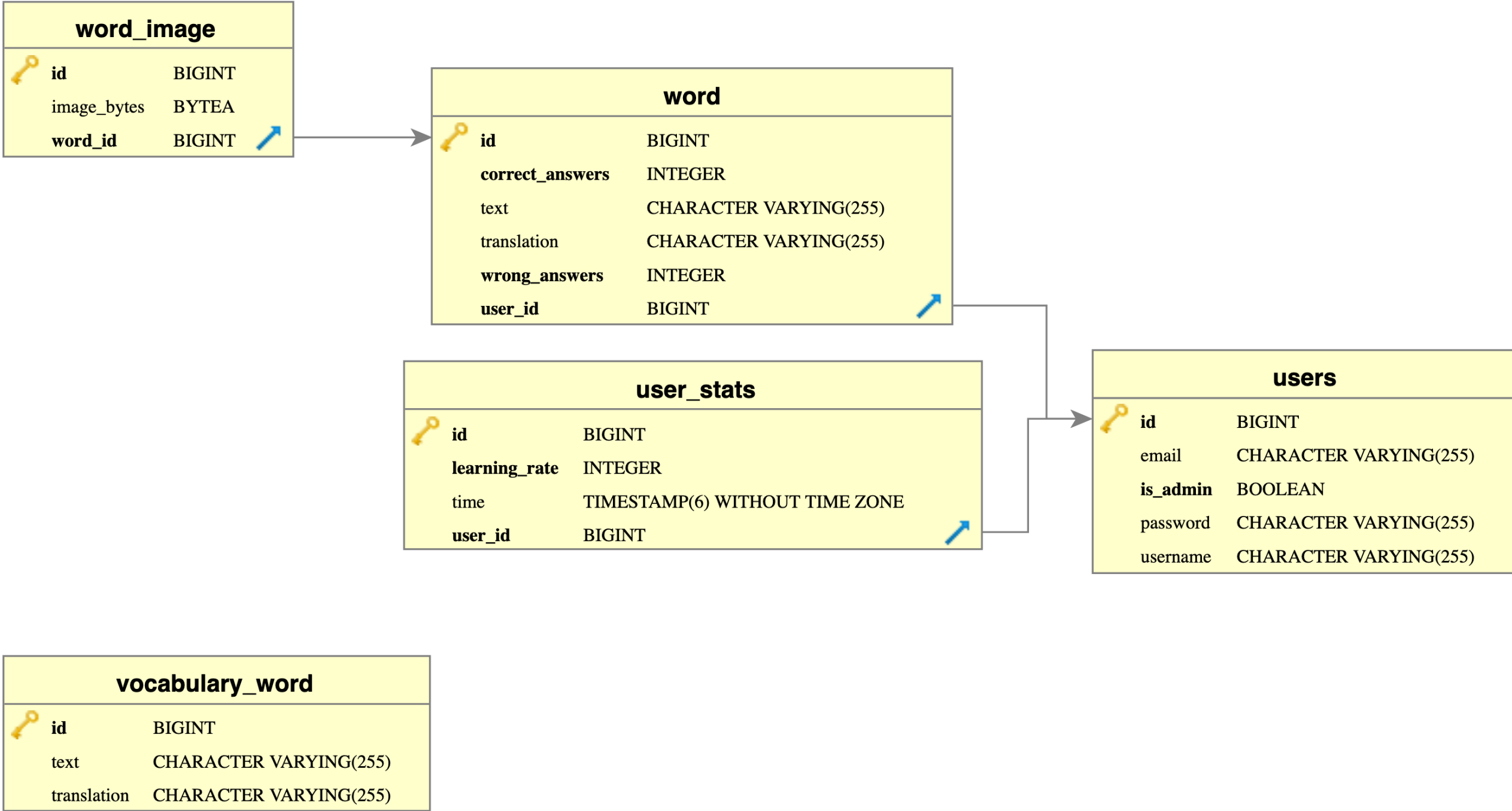
Please repeat following words:

	friends	друзі
	diagram	діаграма

					КПІ.ІП-5123.045440.06.99 КЕ									
					Схема структурна варіантів використання					Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата	Web-застосування для вивчення іноземних мов					Аркуш		Аркушів		
Розробив		Худа А. О.												
Перевірів		Халус О. А												
Т. кон.														
					КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51									
Н. кон.		Ліщук К.І.												
Затвердив		Халус О. А												



					КПІ.ІП-5123.045440.06.99 СС			
					Схема структурна варіантів використання	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Худа А. О.						
Перевірів		Халус О. А.						
Т. кон.					Web-застосування для вивчення іноземних мов	Аркуш		Аркушів
Н. кон.		Ліщук К.І.				КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51		
Затвердив		Халус О. А.						



					КПІ.ІП-5123.045440.06.99 СБД												
					Схема структурна варіантів використання							Літера		Маса	Масштаб		
Зм.	Арк.	№ документа	Підпис	Дата													
Розробив		Худа А. О.															
Перевірив		Халус О. А															
Т. кон.												Аркуш		Аркушів			
					Web-застосування для вивчення іноземних мов							КПІ ім.Ігоря Сікорського Кафедра АСОІУ гр. ІП-51					
Н. кон.		Ліщук К.І.															
Затвердив		Халус О. А															